INSTRUCTIONS :

1. Complete all questions in your designated project group.
2. All members must contribute to writing the codes. (i.e. 1 question = 1 person, and share the workload if there's an additional question relative to the actual number of members in your team (i.e. 5)). Ensure that all members must understand and explain codes from any of the questions.
3. During viva, all students in each team will be randomly asked to describe, answer and edit any of the answers provided. Marks will be given to your ability to present the answers.

Lab Report

Prepare a report to solve the above problems. The report should contain all the sections as below for each question:

Section	Description
1. Problem	Description on the problem
2. Solution	Explanation on how to solve the problems below
3. Sample Input & Output	A few sets of input and output (snapshot)
4. Source code	Java source code

Requirements

1. Group Assignment (Grouping is the same as your project group)
2. Cover page that includes all student matric number and full name.
3. Font: Times New Roman 12, Line Spacing: 1 ½ Spacing
4. Submit to Spectrum according to your OCC. **Deadline** : Before your viva session (W6).

Question 1: Encik Hafiz's Library Fine System

Problem Statement:

Encik Hafiz, the head librarian at the university, is increasingly concerned about overdue library materials. Despite implementing a basic fine structure for "Red Spot" reference books and general items, overdue rates continue to rise, especially among students who frequently borrow materials. To promote accountability and fairness, he has decided to introduce a **dynamic, multi-condition fine system** that penalizes repeat offenders and rewards responsible borrowers.

The new system expands upon the previous one by including more book categories, tiered fines, borrower type considerations, and even historical borrowing behavior. You, the IT department's student assistant, have been tasked with developing this enhanced calculator.

The program must calculate fines based on the **book type**, **number of overdue days**, **borrower type**, and **number of previous late returns**. The fine system must also apply additional penalties for serious overdue cases and discounts for staff or responsible borrowers.

Here are the updated rules:

Book Type & Fine Structure:

- Reference Books (Code 'R'):
  - Strictly for in-library use only. If borrowed and overdue, it incurs a **flat fine of RM 100.00**, regardless of days.

- General Books (Code 'G'):
  - Tier 1 (Days 1–7): RM 0.50 per day
  - Tier 2 (Days 8–30): RM 1.00 per day
  - Tier 3 (Over 30 days): RM 2.00 per day

- Magazines (Code 'M'):
  - Flat rate of RM 0.20 per day.

- Multimedia CDs/DVDs (Code 'C'):
  - RM 2.00 per day for the first 10 days, then RM 5.00 per day afterward.

- Thesis (Code 'T'):
  - RM 10.00 per day flat rate.
  - If overdue **more than 15 days**, a **penalty of RM 200** is applied automatically.

- Example: 65 C S 4
  - Base Fine: (10 days * 2.00) + (55 days * 5.00) = 20.00 + 275.00 = RM 295.00
  - > 60 days penalty: + RM 25.00
  - Habitual offender (4 >= 3): + RM 10.00
  - Total: 295 + 25 + 10 = RM 330.00

Additional Conditions:

- If a book is overdue for **more than 60 days**, an **extra penalty of RM 25.00** is imposed.
- Borrowers with **3 or more previous late returns** are considered “habitual offenders” and receive an **additional RM 10.00 penalty**.
- **Staff (T)** borrowers receive a **20% discount** on the final fine.
- Borrowers with **0 previous late returns** and who return a book **within 3 days late or less** receive a **“Good Borrower Reward”** of **50% fine reduction**.
- Discounts **are mutually exclusive**. If a borrower qualifies for multiple, only one applies. Staff status is checked before Good Borrower status.

The system should prompt the user for:

1. Number of test cases
2. For each case:
    - Days overdue
    - Book type code (R/G/M/C/T)
    - Borrower category (S for Student, T for Staff)
    - Number of previous late returns

The total fine should be displayed with two decimal places.

Sample Input and Output:

```
5
10 G S 1
--- Case 1 ---
Total Fine: RM 6.50

5 M S 0
--- Case 2 ---
Total Fine: RM 1.00

65 C S 4
--- Case 3 ---
Total Fine: RM 330.00

12 T T 2
--- Case 4 ---
Total Fine: RM 96.00

2 G T 0
--- Case 5 ---
Total Fine: RM 0.80
```

Question 2: The Campus "Kopi-Satu" Billing System

Problem Statement:

You have just landed a part-time job at the "Kopi-Satu" café, which is the most popular hangout spot on campus. The lunchtime crowd is hectic, and Kak Sue, the owner, wants to move from her old handwritten receipts to a **smart digital billing system** that automatically calculates taxes, applies discounts, and rewards loyal customers.

The new system not only needs to handle various items with flexible prices but must also apply **conditional taxes**, **tiered discounts**, and **cashback bonuses** depending on the total purchase amount, day of the week, and time of purchase.

Your program should therefore perform the following operations with proper input validation and branching logic:

Requirements:

1. The cashier enters prices for multiple items, one at a time. Entering 0 ends the order.
    - If a **negative price** is entered, display an error and ask for a valid amount again.
    - Ensure at least one valid item is entered before proceeding.

2. Calculate **Service Tax (SST)** based on subtotal:
    - Subtotal ≤ RM 30 → 6%
    - RM 30 < Subtotal ≤ RM 100 → 8%
    - Subtotal > RM 100 → 10%

3. Apply **Discounts**:
    - **Student Saver Discount (Weekdays only):**
      If total before discount > RM 25, apply 10% discount.
    - **Happy Hour Discount (Mon–Fri, 15:00 to 16:59):**
      Additional 5% discount
    - **Weekend Combo Discount (Sat–Sun):**
      If subtotal ≥ RM 50, apply 5% discount.
    - If an invalid day or hour (0-23) is entered, display an error and re-prompt.

4. Apply **Membership Loyalty Cashback:**
    - If the customer enters Y for membership, they earn a **2% cashback** on the final payable amount.

5. Display a detailed receipt showing:
    - Subtotal
    - Service Tax
    - Each discount type applied
    - Final payable amount and cashback (if applicable)

Sample Input 1:

```
Enter item price (0 to finish): 15.00
Enter item price (0 to finish): 22.00
Enter item price (0 to finish): 0
Enter day of week: Friday
Enter hour (24-hour format): 16
Is customer a member (Y/N)? Y
```

Sample Output 1:

```
---- Kopi-Satu Receipt -----------
Subtotal:                 RM 37.00
Service Tax (8%):         RM 2.96
Total before discount:    RM 39.96
Student Discount (10%):   RM 4.00
Happy Hour Discount (5%): RM 1.80
-----------------------------------
Total Payable:            RM 34.17
Loyalty Cashback (2%):    RM 0.68
-----------------------------------
Final Amount to Collect:  RM 34.17
```

Sample Input 2:

```
Enter item price (0 to finish): 4
Enter item price (0 to finish): 2
Enter item price (0 to finish): -1
Invalid amount. Price cannot be negative. Please re-enter.
Enter item price (0 to finish): 0
Enter day of week: Saturday
Enter hour (24-hour format): 22
Is customer a member (Y/N)? N
```

Sample Output 2:

```
---- Kopi-Satu Receipt -----------
Subtotal:                 RM   6.00
Service Tax (6%):         RM   0.36
Total before discount:    RM   6.36
-----------------------------------
Total Payable:            RM   6.36
-----------------------------------
Final Amount to Collect:  RM   6.36
```

Question 3: The Final Year Project Security Check

Problem Statement:

Your friend, Amir, is completing his Final Year Project (FYP), which is a social networking app for university students called "UniVerse." Under the supervision of Dr. Hidayah, a cybersecurity expert, Amir must implement a robust account registration module that ensures **both usernames and passwords** meet strict security standards.

Previously, Amir only focused on password strength. However, Dr. Hidayah's new guidelines require a **two-tier validation process**: first for username integrity, then for password security.

Username Validation Rules:

1. Must **start with a letter** (A–Z or a–z).
2. Must be **5–15 characters** long.
3. Can contain **letters, digits, or underscores** only.
4. Must **not contain uppercase letters** (all lowercase only).
5. If invalid, display "Invalid username" and skip password validation for that case.

Password Strength Rules (7 conditions):

1. Minimum 8 characters
2. At least one uppercase letter
3. At least one lowercase letter
4. At least one digit
5. At least one special character (!, @, #, $, %, ^, &, *)
6. Must not contain any **spaces**
7. Must not contain the **username** as a substring (case-insensitive)

Each password is rated according to how many rules it meets:

Rules Met	Rating
≤ 3	Weak
4–5	Moderate
6	Strong
7	Very Strong

The program should:

- Read a username and password pair
- Validate the username
- If valid, evaluate the password and print its strength

Sample Input 1:

```
Enter username: alice
Enter password: aLicE2024!
```

Sample Output 1:

```
Password Strength: Strong
```

Sample Input 2:

```
Enter username: bob
Enter password: password
```

Sample Output 2:

```
Invalid username
```

Question 4: The Lucky Citizen Challenge

Problem Statement:

During the **Hari Merdeka**, the Malaysian government decides to reward lucky citizens based on their **MyKad (IC) numbers**. Your mission is to **analyze the IC number** and determine whether a citizen qualifies as a **"Lucky Winner."**

Your task:

1. **Extract the birth year, month, and day** from the IC (YYMMDD) and display in full format.
    - Assume: 00–25 → 2000–2025, 26–99 → 1926–1999.
2. **Determine gender** from the last digit:
    - Odd → Male
    - Even → Female
3. **Classify the birth month** as **Long Month** or **Short Month**:
    - **Long Months (31 days):** Jan, Mar, May, Jul, Aug, Oct, Dec
    - **Short Months (28/30 days):** Feb, Apr, Jun, Sep, Nov
4. **Calculate the sum of all digits** in the IC.
5. **Determine "Lucky Winner"** using these rules:
    - **Male:**
        - If **sum of all IC digits is divisible by 5 AND** born in a **Short Month** → Lucky
        - Otherwise → Not Lucky
    - **Female:**
        - If **sum of all IC digits is divisible by 7 AND** born in a **Long Month** → Lucky
        - Otherwise → Not Lucky

Sample input 1:

```
Enter IC number (YYMMDD-##-####): 010203-04-0506
```

Sample output 1:

```
Birth Date: 3/2/2001
Gender: Female
Sum of Digits: 21
Lucky Winner: No
```

Sample input 2:

```
Enter IC number (YYMMDD-##-####): 030831-12-0136
```

Sample output 2:

```
Birth Date: 31/8/2003
Gender: Female
Sum of Digits: 28
Lucky Winner: Yes
```

Question 5: Secret Spy Mission

Problem Statement:

In the distant land of Cipheria, information is the most valuable treasure. The king's secret agents are constantly sending messages about troop movements, treasure locations and magical artifacts. But the neighboring kingdom of Cryptonia has spies everywhere, always trying to intercept and read the messages.

To keep messages safe, the agents use a **special encoding system** based on **binary inversion**:

1. **Every letter** in the original message is first converted into its **ASCII decimal value**.
2. The decimal is then converted into an **8-bit binary**.
3. To confuse enemy spies, the **binary is inverted** — all 0s become 1s and all 1s become 0s.
4. The inverted binary is converted back into **decimal**, forming the **encoded message**.

The king assigns you as a new agent and trusts this system because even if a spy captures the decimal numbers, without knowing the **fixed 8-bit length and inversion rule**, it's almost impossible to decode the original message.

Your task:

1. **Encode the secret message** using the kingdom's inverted-binary system.
2. Print the encoded message as **space-separated decimal numbers**, ready to be transmitted.
3. Ensure that **every binary conversion uses 8 bits**, to prevent errors in the decoding process.

Sample input:

```
Enter Original Message: hello world
```

Sample output:

```
Encoded Message:
151 154 147 147 144 223 136 144 141 147 155
```

Question 6: Dragon Egg Quest

Problem Statement:

In the mystical kingdom of Malaya, you are a young adventurer tasked with finding **magical dragon eggs** hidden in the royal vault. The vault has **10 treasure chests**, numbered from 1 to 10.

Some of the chests are **cursed**. You must find all dragon eggs **before your attempts run out**!

Game rules:

1. Chests and Eggs
    - Total chest: 10
    - Total dragon eggs: 3 (Each egg is hidden in **different** chests include cursed chests **randomly**)
2. Cursed chest
    - There are **2** cursed chests
    - If you guess a cursed chest:
        - You **lose 2 attempts** instead of 1.
        - You see the message: “The chest is cursed! Beware!”
3. Player guesses
    - You guess one chest each time.
    - After each guess:
        - If the chest contains a dragon egg → "You found a dragon egg!"
        - If the chest is empty, you see the messages:
            - Check nearby chest:
                - "Warm! You're very close to a dragon egg!" if within **3 chests** of the **nearest** remaining egg.
                - "Cold! You're far from any dragon egg!" otherwise.
            - Hint on chest number:
                - "Hint: Try a higher chest number." if your guess is less than the nearest remaining egg.
                - "Hint: Try a lower chest number." if your guess is higher than the nearest remaining egg.
            - Always print → "No egg here, keep searching!"
4. Attempts
    - You have a limited number of **10 attempts**.
    - Normal guess uses **1** attempt; cursed chest uses **2** attempts.
5. End Conditions
    - **Win:** You find all dragon eggs → "Congratulations! All dragon eggs are safe!”
    - **Lose:** You run out of attempts → "Game Over! Some dragon eggs remain hidden!"

Sample output:

```
Welcome to the Dragon Egg Quest!
There are 10 chests, 3 dragon eggs, and 2 cursed chests.
You have 10 attempts to find all dragon eggs.

Guess a chest (1-10): 4
Warm! You're very close to a dragon egg!
Hint: Try a higher chest number.
No egg here, keep searching!
Attempts left: 9

Guess a chest (1-10): 6
This chest is cursed! Beware!
Warm! You're very close to a dragon egg!
Hint: Try a higher chest number.
No egg here, keep searching!
Attempts left: 7

Guess a chest (1-10): 7
You found a dragon egg!
Attempts left: 6

Guess a chest (1-10): 2
This chest is cursed! Beware!
You found a dragon egg!
```

```
Attempts left: 4


Guess a chest (1-10): 9
Cold! You're far from any dragon egg!
Hint: Try a lower chest number.
No egg here, keep searching!
Attempts left: 3


Guess a chest (1-10): 3
You found a dragon egg!


Congratulations! All dragon eggs are safe!
```